일변량 자료 탐색

그래프에 의한 일변량 자료 탐색

- 범주형 자료를 위한 그래프
  - 막대 그래프
  - 파이 그래프
  - Cleveland의 점 그래프

- 연속형 자료를 위한 그래프
  - 줄기-잎 그림
  - 상자그림
  - Violin plot
  - 히스토그램
  - 확률밀도함수 그래프
  - 도수분포다각형
  - 점 그래프(dot plot)
  - 경험적 누적분포함수 그래프

1. 범주형 자료를 위한 그래프

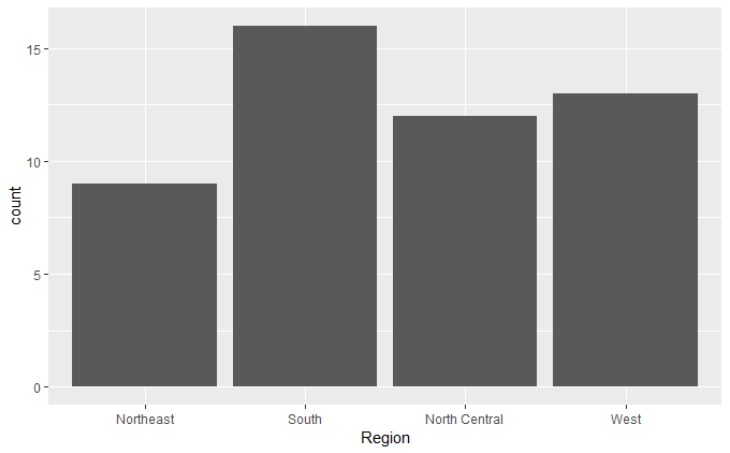
- 막대 그래프
  - 예: state.region

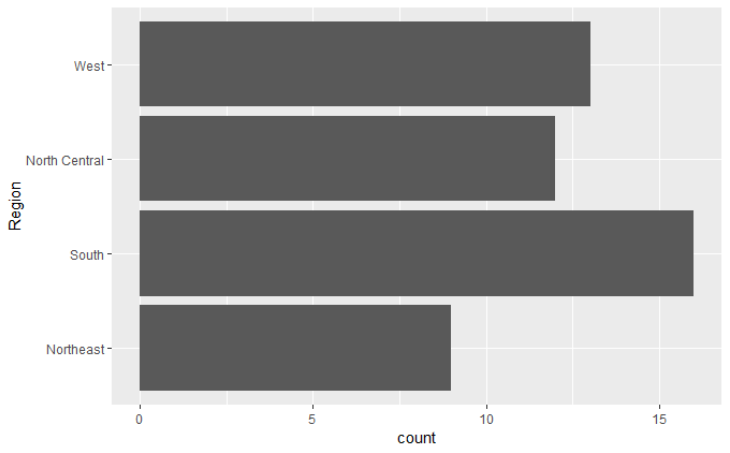
```
> str(state.region)
 Factor w/ 4 levels "Northeast","South",..: 2 4 4 2 4 4 2 2 ...

> state.region[1:5]
[1] South West  West  South West
Levels: Northeast South North Central West
```

1) Input data가 요인인 경우

```
> ggplot(data.frame(state.region)) +
    geom_bar(aes(x=state.region)) +
    labs(x="Region")
```

```
> ggplot(data.frame(state.region)) +
     geom_bar(aes(x=state.region)) +
    labs(x="Region") +
    coord_flip()
```





2) Input data가 도수분포표인 경우

```
> counts <- table(state.region)
> counts
state.region
    Northeast         South North Central          West
            9            16            12            13
```

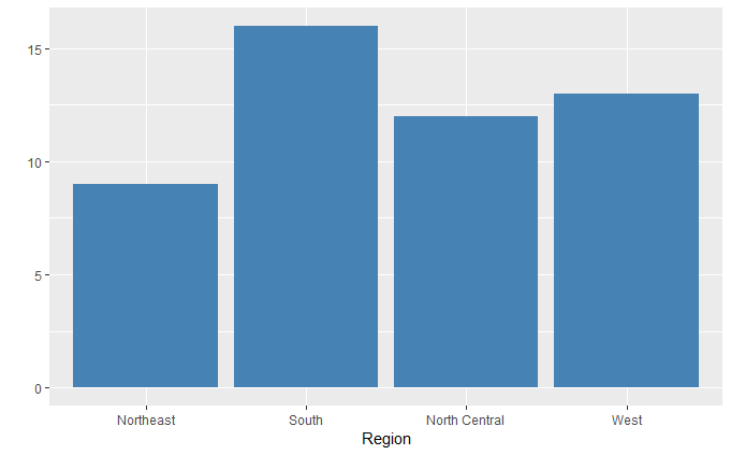
```
> df_1 <- as.data.frame(counts)
> df_1
   state.region Freq
1     Northeast    9
2         South   16
3 North Central   12
4          West   13
```

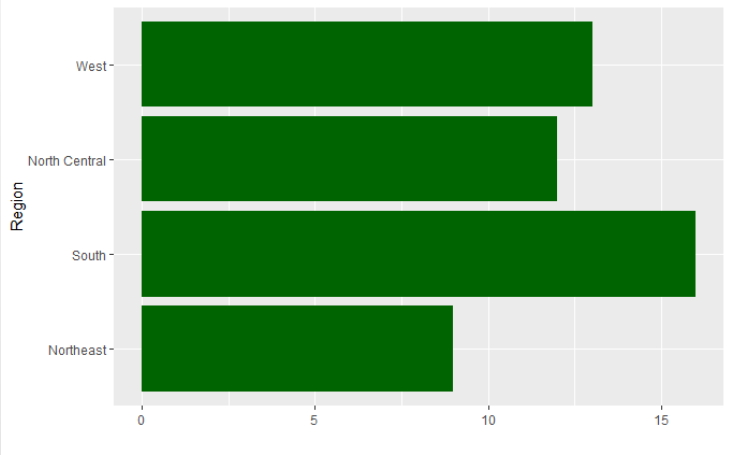
```
> ggplot(df_1,aes(x=state.region, y=Freq)) +
    geom_col(fill="steelblue") +
    labs(x="Region", y="")
```

함수 geom_col( ):

geom_bar(stat="identity")

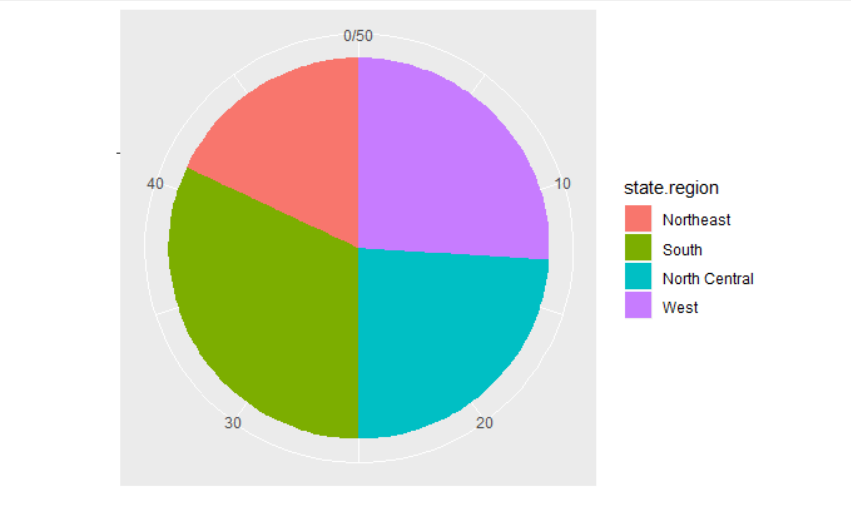
```
> ggplot(df_1,aes(x=state.region, y=Freq)) +
    geom_col(fill="dark green") +
    labs(x="Region", y="") +
    coord_flip()
```



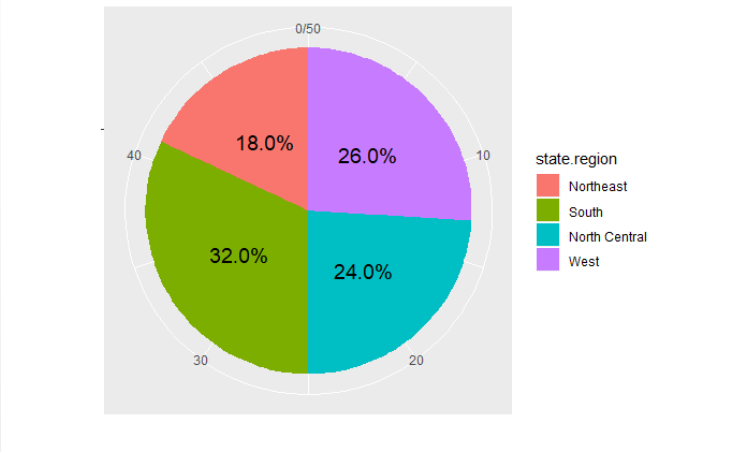


- 파이 그래프
  - 예: state.region

```
> ggplot(data.frame(state.region)) +
    geom_bar(aes(x="", fill=state.region), width=1) +
    labs(x="", y="") +
    coord_polar(theta="y")
```



- 각 파이 조각의 백분율을 라벨로 추가

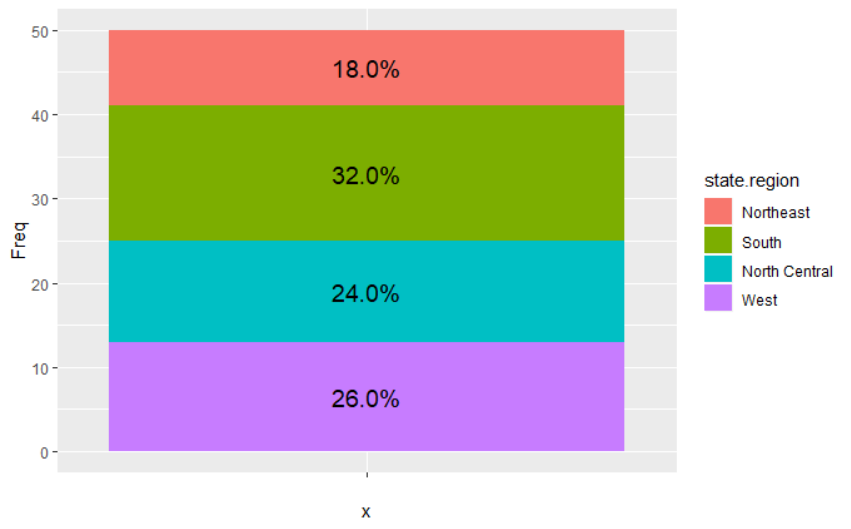


- 패키지 scales의 함수 percent( ): 숫자를 '%' 기호가 있는 백분율로 쉽게 변환

```
> library(scales)
> counts <- table(state.region)
> df_2 <- as.data.frame(counts) %>%
    mutate(pct=percent(Freq/sum(Freq)))
> df_2
   state.region Freq   pct
1     Northeast    9 18.0%
2         South   16 32.0%
3 North Central   12 24.0%
4          West   13 26.0%
```

- 조각의 백분율을 라벨로 추가한 막대 그래프 작성

```
> bar <- df_2 %>%
    ggplot(aes(x="", y=Freq, fill=state.region)) +
    geom_col(width=1) +
    geom_text(aes(label=pct), size=5,
              position=position_stack(vjust=0.5))
> bar
```

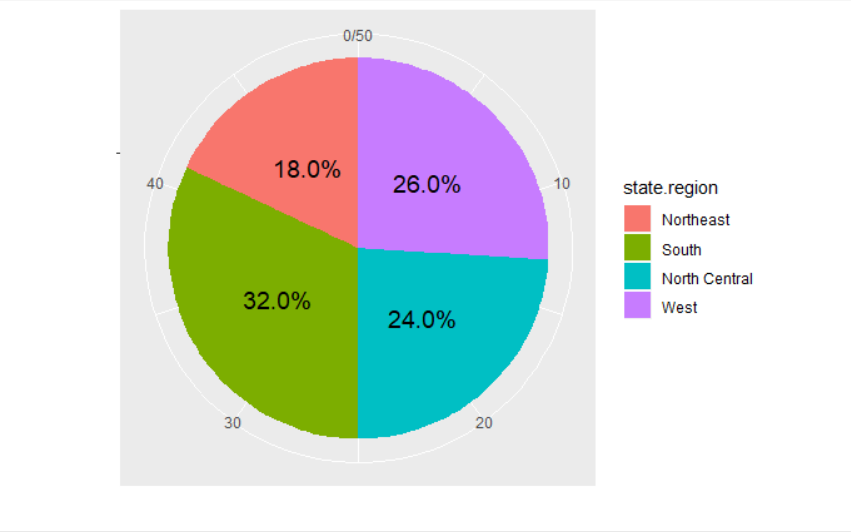


함수 position_stack( ):

- 쌓아 올린 막대 그래프의 각 조각에 라벨 추가 시 매우 유용하게 사용
- 옵션 vjust=0(bottom), 0.5(middle), 1(top; 디폴트)

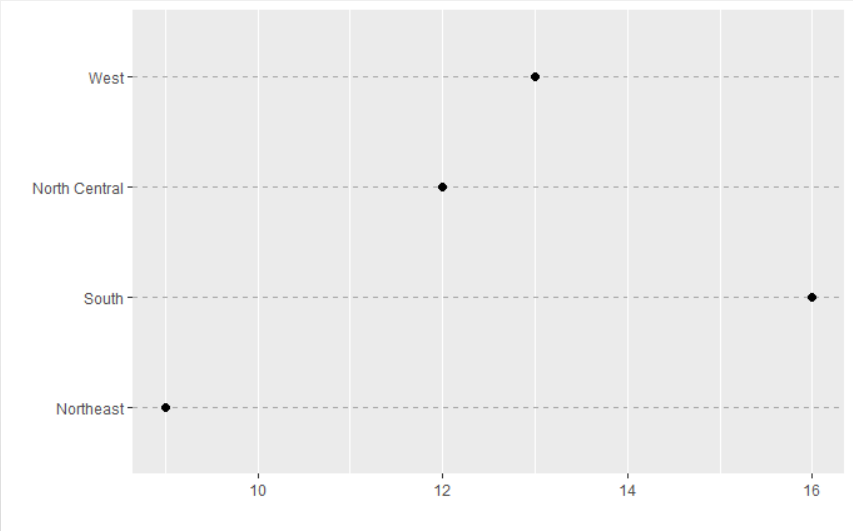
- 조각의 백분율을 라벨로 추가한 막대 그래프를 파이 그래프로 변환

```
> bar + coord_polar("y") +
    labs(x="", y="")
```



- Cleveland의 점 그래프

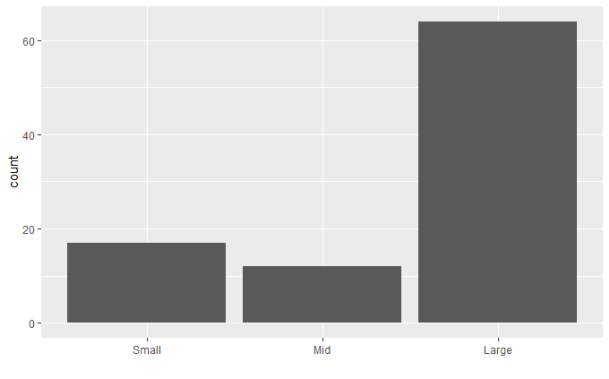
```
> counts <- table(state.region)
> as.data.frame(counts) %>%
    ggplot(aes(x=Freq, y=state.region)) +
    geom_point(size=2) +
    theme(panel.grid.major.y=element_line(linetype=2,
          color="darkgray")) +
    labs(x="", y="")
```

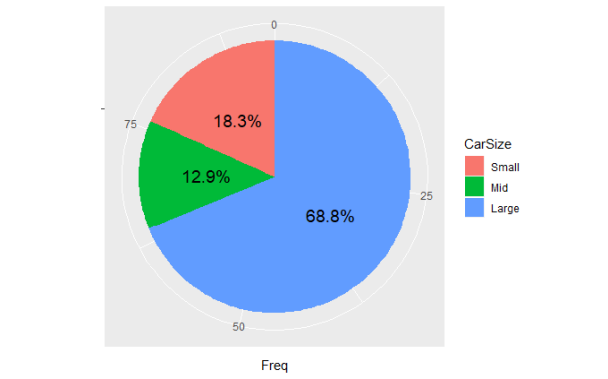


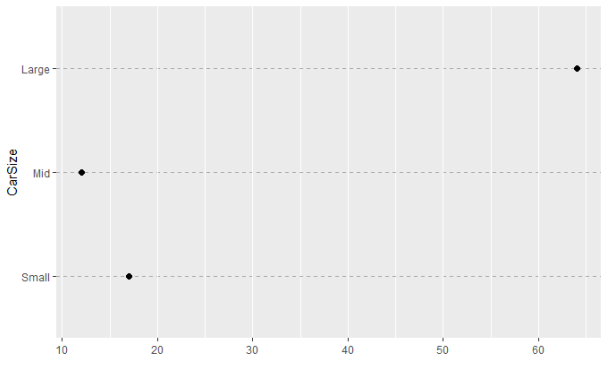
- 연습문제

  - MASS::Cars93

  - 변수 EngineSize를 1.6 이하(Small), 1.6에서 2.0 이하(Mid), 2.0 초과(Large)의 세 범주로 구분하여 각 범주에 대한 도수를 구하고, 이것에 대한 막대 그래프, 파이 그래프, Cleveland의 점 그래프를 다음과 같이 작성하라.







2. 연속형 자료를 위한 그래프

- 줄기-잎 그림
  - 작성법:

    stem(x, scale=1)

    x: 숫자형 벡터

    scale: 그래프의 길이 조절. 줄기의 세분화 정도 조절.

  - 예: women의 변수 height의 줄기-잎 그림

```
> with(women, stem(height))

  The decimal point is 1 digit(s) to the right of the |

  5 | 89
  6 | 01234
  6 | 56789
  7 | 012
```

- 예: 옵션 scale 조절이 필요한 경우

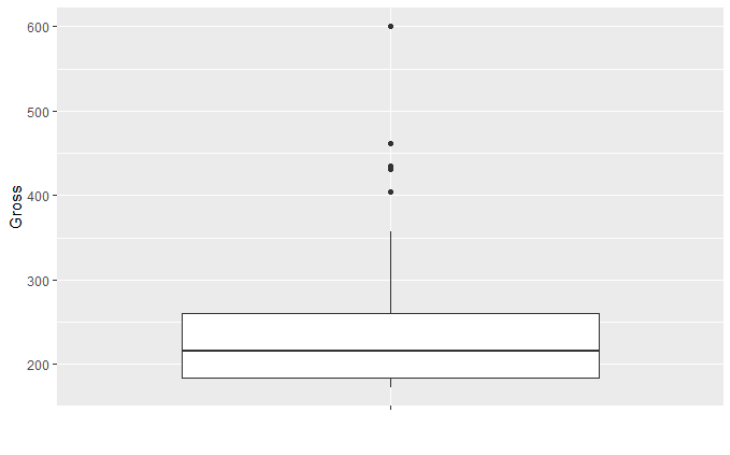
```
> x <- c(98,102,114,122,132,144,106,117,
         151,118,124,115)

> stem(x)

  The decimal point is 1 digit(s) to the right of the |

   8 | 8
  10 | 264578
  12 | 242
  14 | 41
```

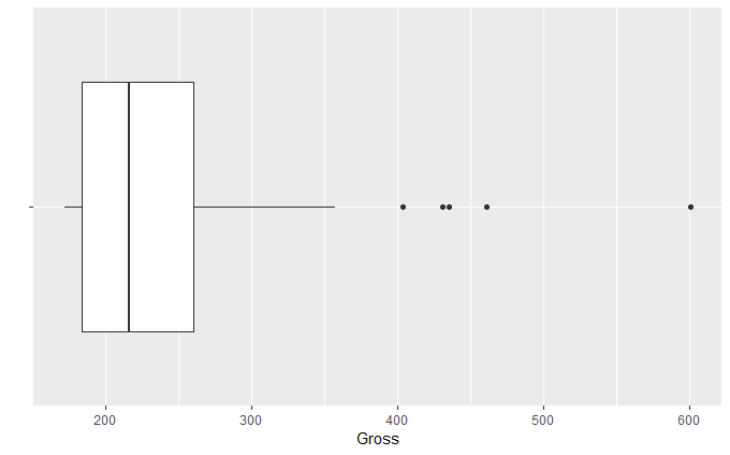
```
> stem(x, scale=2)

  The decimal point is 1 digit(s) to the right of the |

   9 | 8
  10 | 26
  11 | 4578
  12 | 24
  13 | 2
  14 | 4
  15 | 1
```

- 상자그림

  - 예: UsingR::alltime.movies의 변수 Gross의 상자그림 작성

```
> library(UsingR)

> bp <- ggplot(alltime.movies, aes(x="", y=Gross)) +
    geom_boxplot() +
    labs(x="")
> bp
> bp + coord_flip()
```

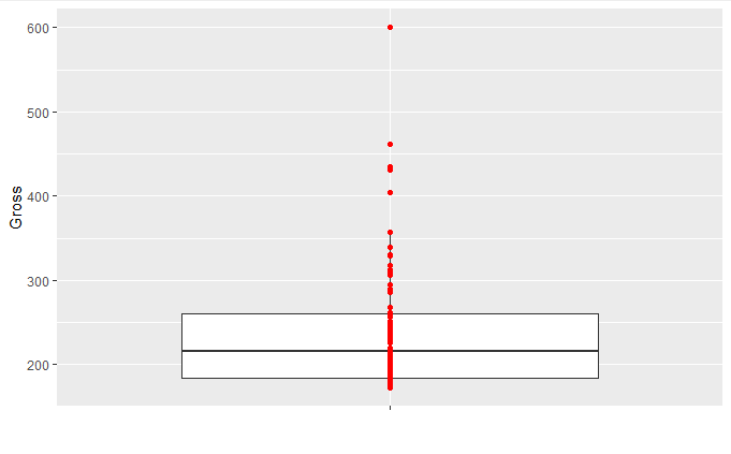




- 상자그림에 자료의 위치를 점으로 표시

  - 함수 geom_point( ) 추가
  - 상자그림에서 이상값을 원으로 표시하는 것 중지: 자료의 점과 겹쳐짐
    outlier.shape=NA 추가

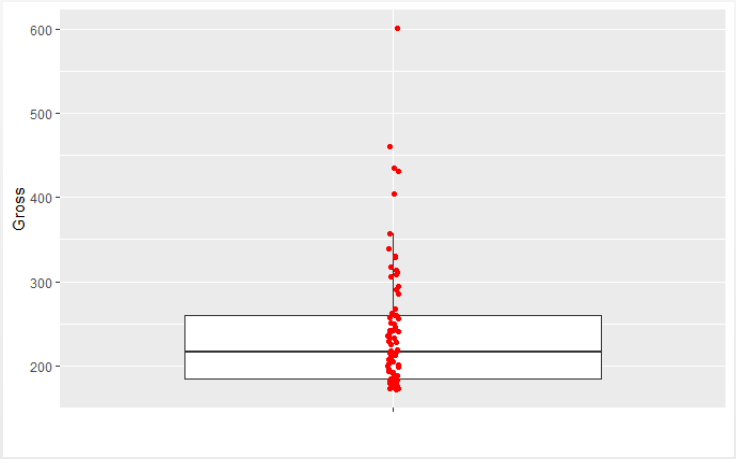
```
> bp1 <- ggplot(alltime.movies, aes(x="", y=Gross)) +
    geom_boxplot(outlier.shape=NA) +
    labs(x="")
> bp1 + geom_point(color="red")
```



- 자료의 점이 겹쳐짐
- geom_jitter( ) 사용이 필요함

- 함수 geom_jitter( )로 상자그림에 자료 위치 표시

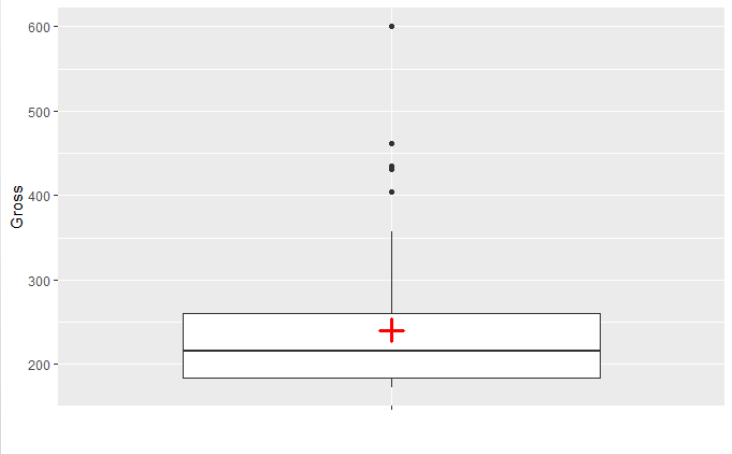
```
> bp1 + geom_jitter(color="red", width=0.01)
```



- 상자그림에 평균값 위치 표시

  - 함수 stat_summary( ): 자료의 요약 통계량을 그래프에 표시
    하나의 x값에 대하여 주어진 y값의 요약 통계량 값 계산
    원하는 요약 통계량: 변수 fun.y에 지정
    원하는 그래프 형태: 변수 geom에 지정

```
> ggplot(alltime.movies,aes(x="", y=Gross)) +
    geom_boxplot() +
    stat_summary(fun.y="mean", geom="point",
                 color="red", shape=3, size=4, stroke=2) +
    labs(x="")
```



- 이상값으로 표시된 자료 확인

  - base graphics 함수인 boxplot( )의 결과물 이용

```
> my_box <- boxplot(alltime.movies$Gross, plot=FALSE)
> my_box$out
[1] 601 461 435 431 404
```

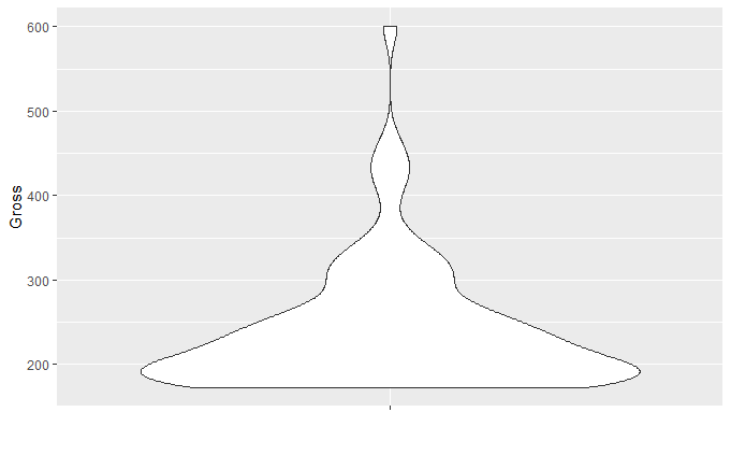
  - 해당 자료 출력

```
> alltime <- as_tibble(alltime.movies) %>%
    rownames_to_column(var="Movie.Title")
> top_movies <- alltime %>%
    filter(Gross %in% my_box$out)
> top_movies
# A tibble: 5 x 3
  Movie.Title                                 Gross Release.Year
  <chr>                                       <dbl>        <dbl>
1 "Titanic                                "   601         1997
2 "Star Wars                              "   461         1977
3 "E.T.                                   "   435         1982
4 "Star Wars: The Phantom Menace          "   431         1999
5 "Spider-Man                             "   404         2002
```

- Violin plot
  - 함수 geom_violin( )으로 작성
  - 확률밀도함수 그래프를 대칭으로 작성한 그래프
  - 상자그림과 겹쳐서 작성하는 것이 일반적 형태

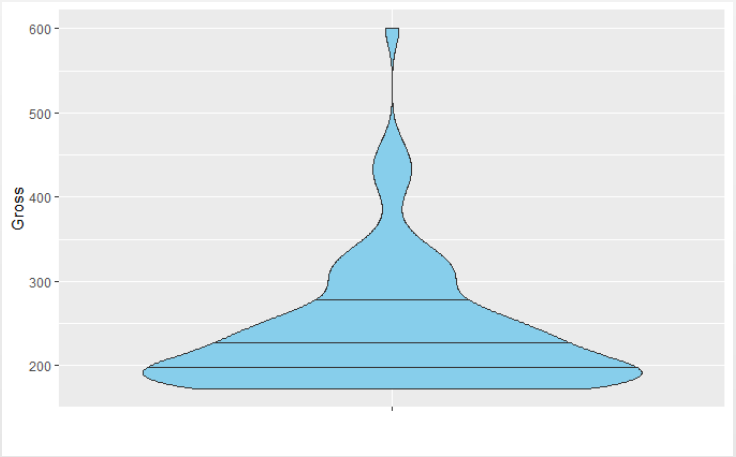
• 예: UsingR::alltime.movies의 변수 Gross의 violin plot 작성

```
> vio <- ggplot(data=alltime.movies, aes(x="", y=Gross)) +
    labs(x="")
> vio + geom_violin()
```



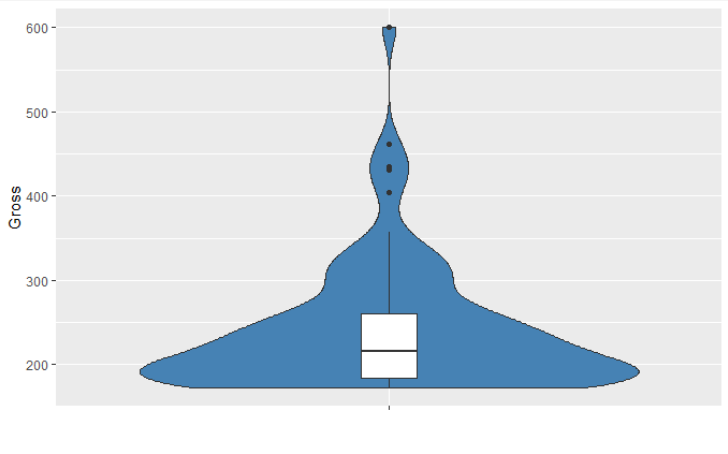
- Violin plot에 분위수 위치 표시
  - 옵션 draw_quantiles에 원하는 분위수 지정

```
> vio + geom_violin(draw_quantiles=c(0.25,0.5,0.75),
                    fill="skyblue")
```



- Violin plot과 상자그림을 함께 작성

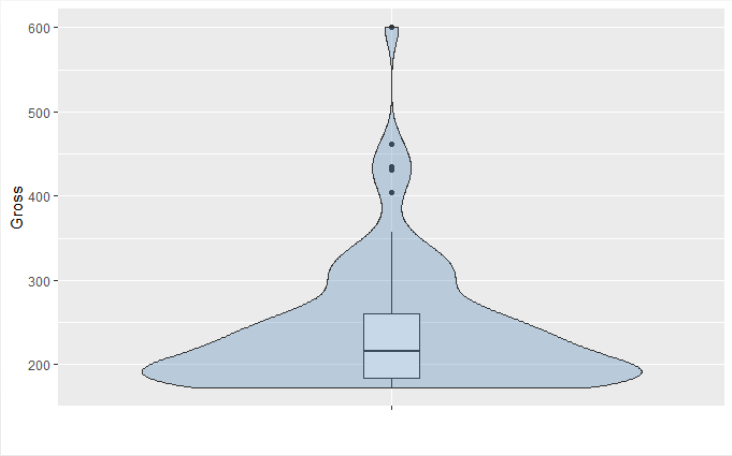
```
> vio + geom_violin(fill="steelblue") +
    geom_boxplot(width=0.1)
```



- 함수 geom_boxplot( )과 geom_violin( )의 순서를 바꾸면 어떤 결과?
- 상자그림이 가려짐
- 대안은 ?

- Violin plot과 상자그림을 함께 작성

```
> vio + geom_boxplot(width=0.1) +
    geom_violin(fill="steelblue", alpha=0.3)
```



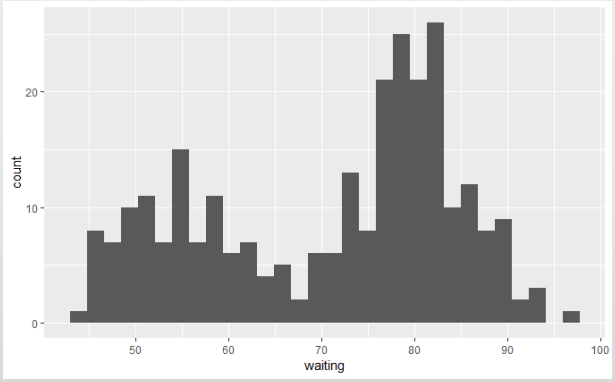
- 변수 alpha로 투명도를 높여서 상자그림을 볼 수 있게 함

- 히스토그램

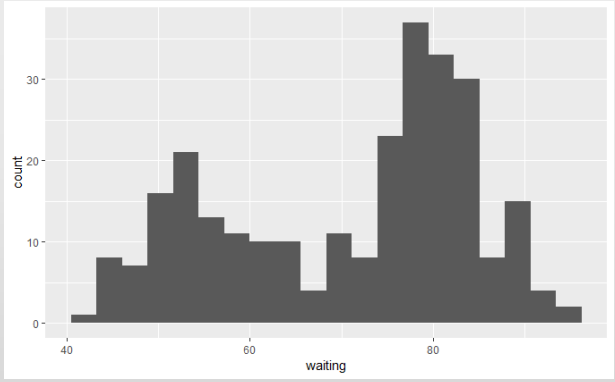
  - 함수 geom_histogram( )
  - 히스토그램의 구간 조절: bins(구간의 개수) 혹은 binwidth(구간 폭)

- 예: faithful의 변수 waiting

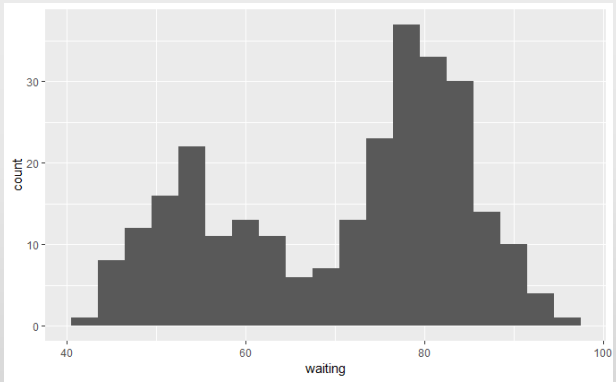
```
> h <- ggplot(faithful, aes(x=waiting))
> h + geom_histogram()
`stat_bin()` using `bins = 30`. Pick better value with `binwidth`.
> h + geom_histogram(bins=20)
> h + geom_histogram(binwidth=3)
```



디폴트 형태



bins=20



binwidth=3

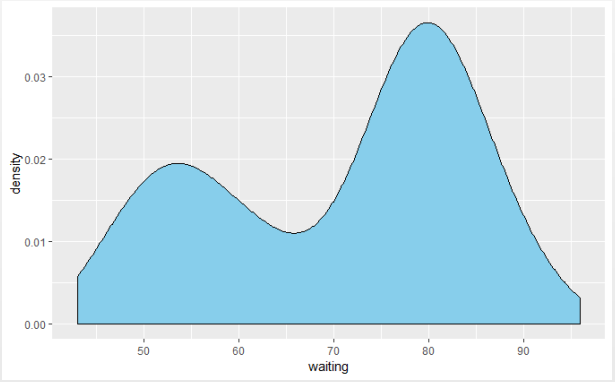
- 확률밀도함수 그래프

  - 연속형 자료의 분포 표현에 가장 적합한 그래프
  - 함수 geom_density( )로 작성

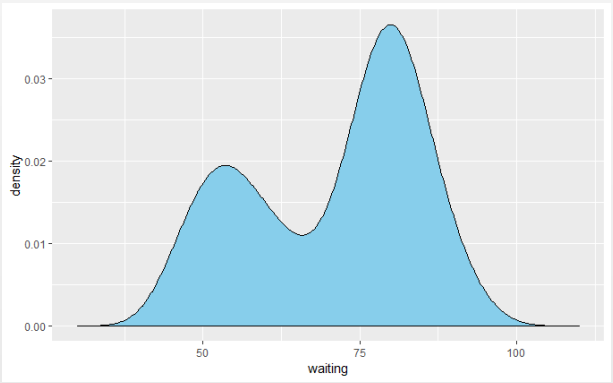
  - 다른 그래프의 문제:
    줄기-잎 그림: 대규모 데이터에는 적합하지 않음
    상자그림: 분포의 세밀한 특징이 나타나지 않음
    히스토그램: 매끄럽지 않은 계단함수의 형태

- 예: faithful의 waiting

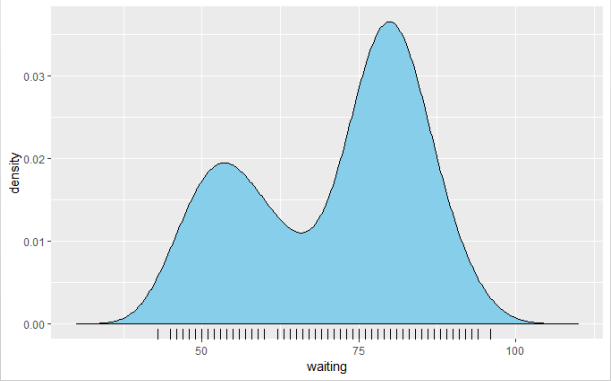
```
> p <- ggplot(faithful, aes(x=waiting)) +
    geom_density(fill="skyblue")
> p
> p + xlim(30,110)
> p + geom_rug() + xlim(30,110)
```



디폴트 형태



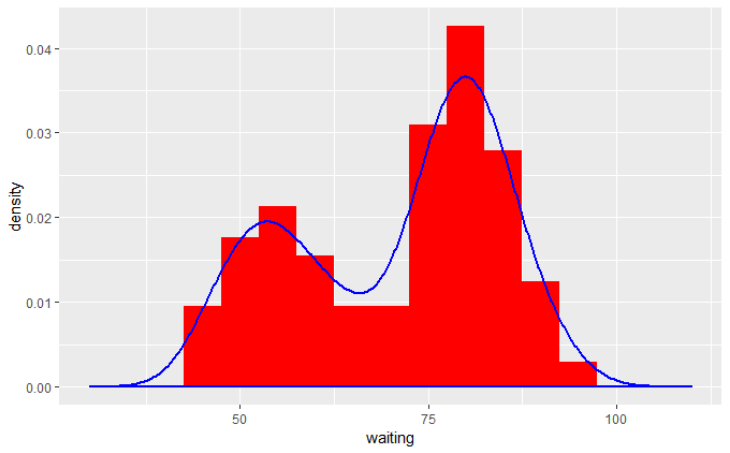
x축 구간 확대
함수 xlim( ) 사용



자료 위치 추가
함수 geom_rug( ) 사용

- 히스토그램과 겹치게 작성

```
> ggplot(faithful, aes(x=waiting, y=..density..)) +
    geom_histogram(fill="red", binwidth=5) +
    geom_density(color="blue", size=1) +
    xlim(30,110)
```



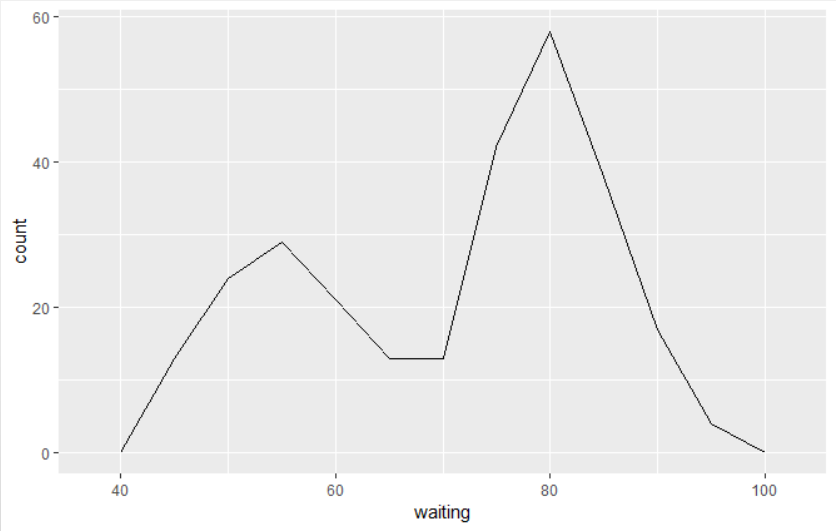
함수 geom_density( )와
geom_histogram( )의 실행 순서를 바꾸면?

- 도수분포다각형(Frequency polygon)
  - 히스토그램: 각 구간에 속한 자료의 도수를 높이로 하는 막대
  - 도수분포다각형: 각 구간의 도수를 선으로 연결
  - 작성 geom 함수: geom_freqpoly( )
    사용법은 geom_histogram( )과 동일

- 예제: 데이터 프레임 faithful의 waiting에 대한 도수분포다각형

```
> pp <- ggplot(faithful, aes(x=waiting))
> pp + geom_freqpoly(binwidth=5)
> pp + geom_freqpoly(aes(y=..density..), binwidth=5)
```

도수분포다각형



상대도수분포다각형